2-15-2013

Data mining the functional characterizations of proteins to predict their cancer-relatedness

Peter Revesz
University of Nebraska-Lincoln, prevesz1@unl.edu

Christopher Assi
University of Nebraska-Lincoln, cassi@cse.unl.edu

Data Mining the Functional Characterizations of Proteins to Predict their Cancer-Relatedness

Peter Z. Revesz, Christopher J.-L. Assi

***Abstract*—**This paper considers two types of protein data. First, data about protein function described in a number of ways, such as, GO terms and PFAM families. Second, data about whether individual proteins are experimentally associated with cancer by an anomalous elevation or lowering of their expressions within cancerous cells. We combine these two types of protein data and test whether the first type of data, that is, the functional descriptors, can predict the second type of data, that is, cancer-relatedness. By using data mining and machine learning, we derive a classifier algorithm that using only GO term and PFAM family descriptions of a protein can predict with over 73 percent accuracy whether it is associated with pancreatic cancer.

***Keywords*—**data mining, GO term, pancreatic cancer, PFAM family, protein.

I. Introduction

Data mining is increasingly applied to non-relational databases, including genome and protein databases [1]--[5]. Data mining and data classification methods are developed for protein structure and function [6]--[8], protein evolution [9], [10], protein interaction networks [11], and medical data that may include genomes or proteins [12]--[14]. In the present paper, preliminary versions of which were presented in [15] and [16], we focus on a *pancreatic cancer protein database*. This database was collected by Robert Powers and Bradley Worley, in the Department of Chemistry at the University of Nebraska-Lincoln, based on earlier pancreatic cancer research [17]--[23].

Pancreatic cancer was chosen as a test case because it has the lowest survival rate among different types of cancer. Data mining was used to investigate the relationship among anomalous proteins, which have unusually high or low levels in pancreatic patients. Early recognition of some patterns developing among these anomalous proteins may allow treatment to start earlier and increase the survival rate of pancreatic cancer patients.

Data mining of protein databases poses special challenges because many protein databases often contain set data types, whereas most data mining and machine learning algorithms assume relational database inputs. We overcame this problem by describing effecting ways to restructure the protein databases into relational databases. The restructured databases allowed the use of several types of classifiers, such as, Support Vector Machines (SVMs) and decision trees. Other types of data mining algorithms could be also used, but we chose these two types because they are currently the most frequently used data mining methods.

This paper is organized as follows. Section II describes some basic background. Section III presents the experimental results of applying the J48 decision tree and the libSVM classifiers to the restructured pancreatic cancer database. Section IV gives a detailed analysis of the prostaglandin protein synthesis network. Section V discusses the results. Finally, Section VI gives our conclusions and possible directions for future work.

P. Z. Revesz is with the Department of Computer Science and Engineering, University of Nebraska-Lincoln., Lincoln, NE 68588 USA (phone: 571-201-5639; fax: 402-472-7767; e-mail: revesz@cse.unl.edu).

C. J.-L. Assi is with the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588 USA (e-mail: cassi@cse.unl.edu).

II. Background Concepts and Tools

In this section, part A gives an introduction to classifiers and part B describes the WEKA system that contains a library of implemented classifiers.

A. Classifiers

Let $R(x_1,\ldots,x_n, y)$ be a relation, where the set of attributes $X = \{x_1,\ldots,x_n\}$ is called the *feature space* and the y attribute is called a *label*. Each tuple of the relation describes some entity based on specific values of the feature space and the label. For example, each row may describe a protein with specific feature attributes, such as, molecular weight, amino acid sequence etc., and a label attribute, such as, whether it is involved in pancreatic cancer.

Given such a relation R, a classifier is mapping from X to y. If a classifier is correct on all tuples of relation R, then the value of y can be always predicted from the values of X. In practice, the classifier may not be correct on all proteins. Further, classifiers are intended to be able to classify even those proteins that are new, not just those that are already in R. Popular classifiers include *decision trees* and *Support Vector Machines* (*SVM*s). A decision tree is a tree which is read from the root towards the leaves, and whose internal nodes are tests and whose leaf nodes are categories [24]. For example, C4.5 is a well-known decision tree algorithm [25]. SVMs perform classification by constructing for relation R an n-dimensional hyperplane that optimally separates the data into two categories (for example when $y = 0$ and $y = 1$). An example of

SVM is the libSVM implementation [26].

B. The WEKA Library

In our experiments we used the *Waikato Environment for Knowledge Analysis* (*WEKA*) system developed at the University of Waikato [27], [28]. WEKA provides an extensive library of data mining and machine learning algorithms. In WEKA, the input data is a relation or table which is represented by an *Attributes Relation File Format* (*ARFF*) file. Each ARFF file starts with a title to let the user know what kind of data is stored in the file. The title is followed by a relation type and then all the attributes and their types. Finally, the attribute declarations are followed by the actual data rows.

C. The Restructuring Method in Theory

In the pancreatic protein database collection of about eighty tables, we chose for our study the GO_np and PFAM_np tables, which contain data about pancreatic proteins that are not involved in cancer, and the GO_pdac and PFAM_pdac tables, which contain data about pancreatic proteins that are related to pancreatic cancer. GO_np had 70,331, PFAM_np had 7,054, GO_pdac had 30,888, and PFAM_pdac had 7,272 rows, that is, a total number of 115,545 rows. A simplified version of the GO_pdac looks as follows:

Table 1 The GO_pdac table.

UID	GO
O43491	GO:0003779
O43491	GO:0005198
O43491	GO:0005886
O43491	GO:0008091
O43491	GO:0019898
O43491	GO:0030866
Q96C24	GO:0005215
Q96C24	GO:0005886
Q96C24	GO:0019898
Q96C24	GO:0030658
Q96C24	GO:0042043
…	…

The GO_pdac table lists all (UID, GO) pairs, such that UID is the universal identifier of a pancreatic protein and GO is a feature descriptor, also called a GO term. The UID and the GO terms can be found in the UNIPROT database (www.uniprot.org). There is a many-to-many relationship between the UIDs and the GO terms. For example, rows three and five with the same UID O43491 are related to two different GO terms, GO:0005886 and GO:0019898. On the other hand, rows three and eight with the same GO term GO:0005886 are related to two different UIDs, O43491 and Q96C24.

The GO_np tables listed (UID, GO) pairs of non-pancreatic proteins. We merged the GO_np and GO_pdac tables without losing the information whether the protein is related to cancer or not. Hence we extended the GO_np and the GO_pdac tables with a Y column, which denotes whether the protein is related to pancreatic cancer or not. All the proteins in the GO_np table are extended with a Y value of "0", while all the proteins in the GO_pdac table are extended with a Y value of "1" by the following SQL query, which we call SQL 1 in Fig. 1:

```
create view GO_merge (UID, GO, Y) as
select UID, GO, 0 from GO_np
union
select UID, GO, 1 from GO_pdac;
```

After the above query is executed the GO_merge table looks as follows:

Table 2 The GO_merge table.

UID	GO	Y
O43491	GO:0003779	1
O43491	GO:0005198	1
O43491	GO:0005886	1
O43491	GO:0008091	1
O43491	GO:0019898	1
O43491	GO:0030866	1
Q96C24	GO:0005215	1
Q96C24	GO:0005886	1
Q96C24	GO:0019898	1
Q96C24	GO:0030658	1
Q96C24	GO:0042043	1
…	…	…

We restructured or "*flattened*" the above table by an SQL query that transformed GO_merge into another table GO_merge_flat in which all information about a single protein appears in one row, as shown in Table 3.

Table 3 The GO_merge_flat table.

UID	0003779	0005198	0005886	0008091	0019898	0030866	0030658	Y
O43491	1	1	0	1	1	1	1	1
Q96C24	0	0	1	1	0	1	0	1

In theory, the number of attributes in the restructured relation is n+2, where n is the number of distinct GO terms. Apart from UID and Y, these distinct GO terms form the attributes of the restructured relation. Below each GO term a '1' or '0' indicates whether the GO term applies to the protein indicated by the UID on the left.

D. Simplifying the Restructuring Problem

The restructuring method described in the part C is not practical because it requires a huge matrix. For example, since GO_merge table contains 17943 distinct UIDs and 7935 distinct GO terms, a straightforward application of the restructuring method would yield a table with

$$17943 \times (7935 + 2) \approx 1.6 \times 10^8$$

entries. The WEKA and other machine learning systems simply cannot handle such big matrices. Moreover, the matrix could become even bigger when we consider not only GO terms but PFAM families and other attributes as described in part E below.

One possible way to reduce the size of the matrix is using Principal Component Analysis. Using Principal Component Analysis, the matrix could be rewritten into another matrix with a smaller number of columns. The new columns would be linear combinations of the existing columns, that is, the 7935 GO terms. While this would reduce the size of the matrix and alleviate the runtime problems with WEKA and other machine learning systems, it would still not be a good solution.

Our ultimate goal is to be able to easily and accurately identify whether a new protein may be associated with cancer. Intuitively, we would like to characterize the cancer-related proteins based only on a *small subset of the GO terms* because it is impractical to test each of the 7935 GO terms whether it applies to a new protein. The Principal Component Analysis would still require that we test each of the 7935 GO terms, and then linearly combine their (1 or 0) values to find the new columns. That is why the Principal Component Analysis would not yield a satisfying solution.

We need another method to find a *small subset of the GO terms* that characterizes the proteins in terms of cancer-relatedness as accurately as the entire set of GO terms would characterize those. How can we find such a subset of the GO terms?

Luckily, the restructuring matrix would be very sparse because most of the UIDs are characterized by less than ten GO terms. Hence most of the 7935 distinct GO terms would have a value of 0 in most rows. GO terms that occur only rarely do not connect many different UIDs hence they are not very useful as efficient cancer indicators.

Hence we experimented with selecting only the top *n* most frequent GO terms. We observed that in general when *n* increases the accuracy also increases. At some point the increase in the accuracy diminishes with further increments in *n*. Hence it is not worth to increase further the value of *n* beyond that point. In our case, this value of *n* was about 200.

Therefore, in the experiments presented below we selected only the top 200 most frequent GO terms as follows. First we found the frequency of each Go terms using the following SQL query:

```
create view GOcount(GO,count) as
select GO, count(*)
from GO_merge
group by GO;
```

The new table GOcount(GO,count) contains the count of each GO term. We extracted the top 200 most frequent GO terms into a text file as follows:

```
select GO from GOcount
order by count desc limit 200
into outfile `/tmp/MergeTop200GO.txt';
```

We wrote a C++ program, which is shown in detail in the APPENDIX, to automatically generate the restructuring SQL query. Apart from some initialization and ending, the program repeatedly reads the next GO term from the input file MergeTop200GO.txt and writes to an output file SQL_flatten.txt the line of the SQL query that corresponds to the GO term. Below is how the SQL_flatten.txt file looks like.

```
select UID,
  max(case when GO = `GO:0016021' then 1 else 0 end) as
`GO:0016021',
  max(case when GO = `GO:0005515' then 1 else 0 end) as
`GO:0005515',
  max(case when GO = `GO:0005634' then 1 else 0 end) as
`GO:0005634',
  max(case when GO = `GO:0005737' then 1 else 0 end) as
`GO:0005737',
  max(case when GO = `GO:0008270' then 1 else 0 end) as
`GO:0008270',
  max(case when GO = `GO:0006350' then 1 else 0 end) as
`GO:0006350',
  max(case when GO = `GO:0007165' then 1 else 0 end) as
`GO:0007165',
  max(case when GO = `GO:0005886' then 1 else 0 end) as
`GO:0005886',
  max(case when GO = `GO:0005524' then 1 else 0 end) as
`GO:0005524',
  max(case when GO = `GO:0003677' then 1 else 0 end) as
`GO:0003677',
  …
  Y
  from GO_merge
  group by UID
```

When the above SQL query is executed, for each UID it checks all the GO terms. If any of the GO terms the UID is associated with matches a particular GO term for which we are creating a column in the flattened table, then that GO term will get a value of ``1" else it will get a value of ``0". The process then continues until it does not read any more UID groups.

E. Merging GO_merge and PFAM_merge

The PFAM table is similar to the GO table. The PFAM table contains the UID of proteins and the PFAM terms, which form another set of characterizations of proteins as an alternative to the GO term characterization. We can create PFAM_merge by merging PFAM_np and PFAM_pdac similarly to how we created GO_merge. Fig. 1 outlines the process of merging the GO_merge and the PFAM_merge tables together when we need to use both the GO and the PFAM terms. Table 4 is an example PFAM_merge table. The SQL query, called SQL 2 in Fig. 1, to generate the PFAM_merge table is similar to the SQL 1 query we saw before.

In Fig. 1, SQL 3 refers to the following query:

```
select T.UID,
max(case when GO = `GO:0016021' then 1 else 0 end) as
`GO:0016021',
…
max(case when family = `PF07647' then 1 else 0 end) as
`PF07647'
…
, T.Y
from GO_merge T JOIN PFAM_merge ON T.UID =
PFAM_merge.UID
group by UID
```

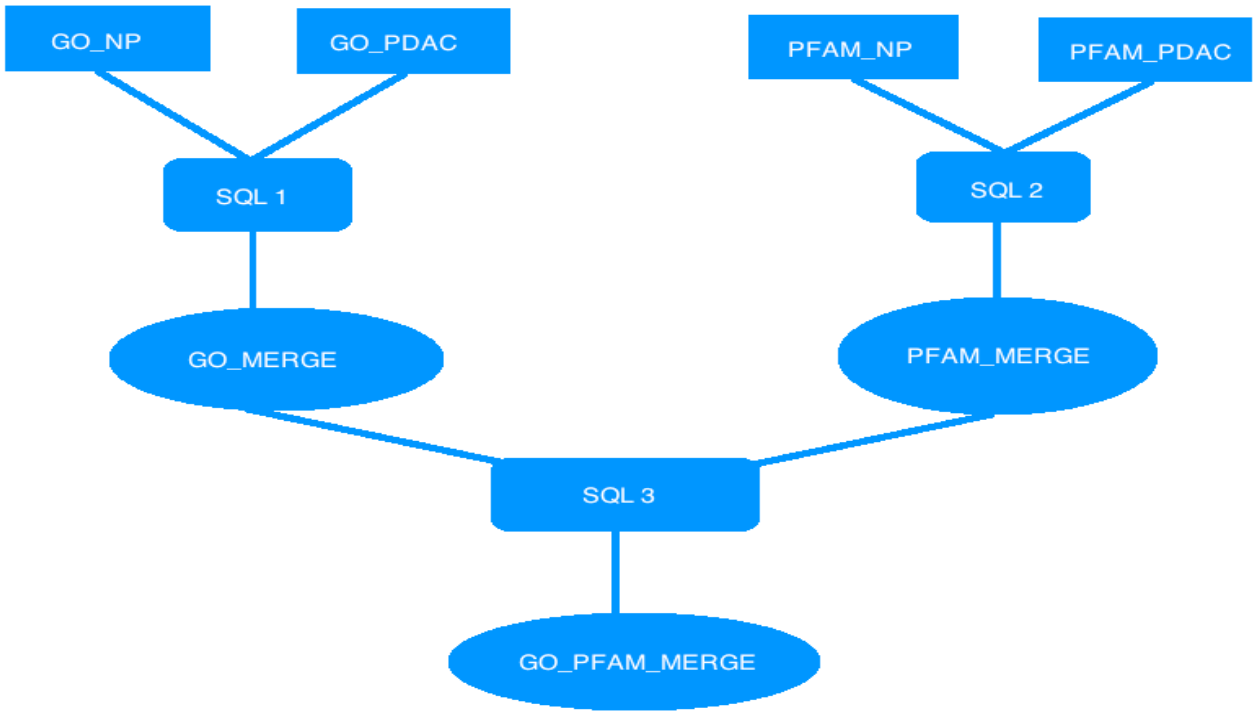

Fig. 1 Generating GO_PFAM_merge.

Table 4 The PFAM_merge table.

UID	PFAM	Y
P02656	PF05778	0
P09651	PF00076	0
Q9BY79	PF00431	0
Q9BY79	PF01392	0
Q9BY79	PF00057	0
O95931	PF00385	0
Q9UKU0	PF00501	0
P10323	PF00089	0
Q17RR3	PF00151	0
Q17RR3	PF01477	0
…	…	…

In our experiments, we used the top n most frequent GO terms as well as the top m most frequent PFAM terms, yielding a relation with $n+m+2$ attributes. We varied the values of n and m as described in the next section.

III. Experimental Results

Given a flattened file, as in Table 3, it is easy to generate an ARFF file, which is needed for the WEKA system. In the ARFF file, the UID attribute ranges over strings that describe protein IDs, and the "relation" attribute substitutes for the "Y" attribute. For example, Table 3 is described using ARFF as follows:

```
@relation GO_merge_flat
@attribute ``UID" {O43491, Q96C24}
@attribute ``GO:0003779” {0, 1}
@attribute ``GO:0005198” {0, 1}
@attribute ``GO:0005215” {0, 1}
@attribute ``GO:0005886” {0, 1}
@attribute ``GO:0008091” {0, 1}
@attribute ``GO:0019898” {0, 1}
@attribute ``GO:0030866” {0, 1}
@attribute ``relation" {0, 1}
@data
``O43491",1,1,0,1,1,1,1,1
``Q96C24",0,0,1,1,0,1,0,1
```

From our WEKA library, we used the libSVM support vector machine, which was previously added to the library, and the J48 decision tree. Both of these accepted input in ARFF format. The stratified cross-validation was used in all our classifications.

A. *Support Vector Machine Results*

Using libSVM with the GO_merge_flat file, WEKA gave the following:

CORRECTLY CLASSIFIED:	12947	72.156 %
INCORRECTLY CLASSIFIED:	4996	27.844 %
TOTAL NUMBER:	17943	100 %

WEKA also gave the following confusion matrix:

a	b	CLASSIFIED
12794	305	a = 0
4691	153	b = 1

The confusion matrix displays the relationship between two or more categorical variables. The number of correctly classified instances is the sum of the diagonals in the confusion matrix; all the others are incorrectly classified. For libSVM with the PFAM_merge file and stratified cross-validation, the data mining results with were as follows:

CORRECTLY CLASSIFIED:	11590	71.707 %
INCORRECTLY CLASSIFIED:	4573	28.293 %
TOTAL NUMBER:	16163	100 %

The classification for all our instance was correct for about 71.7 % of the instances. Below is the confusion matrix:

a	b	CLASSIFIED
163	4263	a = 0
310	11427	b = 1

B. *Decision Tree Results*

Our next set of experiments used the J48 decision tree. The decision tree with the GO_merge_flat file gave the following results:

CORRECTLY CLASSIFIED:	12922	72.017 %
INCORRECTLY CLASSIFIED:	5021	27.983 %
TOTAL NUMBER:	17943	100 %

The classification was again about 72 % correct. Below is the confusion matrix for the J48 decision tree:

a	b	CLASSIFIED
12562	537	a = 0
4484	360	b = 1

For decision tree with the PFAM_merge_flat file, the data mining results were as follows:

CORRECTLY CLASSIFIED:	11719	72.505 %
INCORRECTLY CLASSIFIED:	4444	27.495 %
TOTAL NUMBER:	16163	100 %

The classification for all our instances was correct for over 72 % of the instances. It was slightly better than for GO_merge_flat with the decision tree classification. Below is the confusion matrix for the PFAM_merge decision tree:

a	b	CLASSIFIED
144	4282	a = 0
162	11575	b = 1

C. Improving the Accuracy

As we saw above, for both the GO_merge_flat and the PFAM_merge_flat files and both the libSVM and the J48 the accuracy was around 72 %. A natural question is whether the accuracy can be improved by using both the GO terms and the PFAM families together. As we saw in Fig. 1, these terms can be combined in a relation GO_PFAM_merge. This file can be also flattened and represented in ARFF. We performed another set of experiments using WEKA and the GO_PFAM_merge_flat file. The results for the libSVM were the following:

CORRECTLY CLASSIFIED:	13099	73.003 %
INCORRECTLY CLASSIFIED:	4844	26.997 %
TOTAL NUMBER:	17943	100 %

Finally, the results for J48 were the following:

CORRECTLY CLASSIFIED:	12936	72.095 %
INCORRECTLY CLASSIFIED:	5007	27.905 %
TOTAL NUMBER:	17943	100 %

Our results from the GO_PFAM_merge analysis show that the libSVM has the highest percentage of 73 % compared to 72 % for the decision tree.

IV. PROSTAGLANDIN SYNTHESIS

Several recent studies have identified prostaglandin to be a major factor in pancreatic cancer [29]-[31]. We retrieved from the UNIPROT database (www.uniprot.org) all prostaglandin-related proteins using the following query:

(prostaglandin AND organism:"Homo sapiens [9606]")
AND reviewed:yes

The query retrieved 89 proteins, but many of those were indicated to belong specifically to the liver, brain or other organs. By cross-checking with our pancreatic protein database, we identified the 24 pancreatic and prostaglandin-related proteins shown in Table 5. The prostaglandin-related proteins interact with each other as shown in Fig. 2.

Combining all of the information in Table 5 and Fig. 2, we hypothesize that in pancreatic cancer the following *chain of events* takes place, where "anomaly" means either over-expressed or under-expressed.

Table 5. Prostaglandin-related pancreatic proteins.

UID	Name	Y
O15496	Group 10 secretory phospholipase A2	0
O60733	Group VI phospholipase A2	0
O60760	Glutathione S-transferase	0
P02775	Platelet basic protein	1
P04083	Phospholipase A2 inhibitory protein	1
P08047	Transcription factor Sp1	1
P09917	Arachidonate 5-lipoxygenase (ALOX5)	1
P23219	Prostaglandin H2 synthase 1(COX-1)	1
P24557	Thromboxane-A synthase	0
P34995	Prostaglandin E2 receptor EP1 subtype	0
P35354	Prostaglandin G/H synthase 2 (COX-2)	0
P35408	Prostaglandin E2 receptor EP4 subtype	1
P41222	Prostaglandin D2 synthase	1
P43115	Prostaglandin E2 receptor EP3 subtype	1
P43116	Prostaglandin E2 receptor EP2 subtype	1
P47712	Cytosolic phospholipase A2	0
Q14684	Prostaglandin E synthase	0
Q15185	Cytosolic prostaglandin E2 synthase	1
Q16647	Prostacyclin synthase	0
Q68DD2	Cytosolic phospholipase A2 zeta	0
Q92959	Prostaglandin transporter	1
Q9H7Z7	Prostaglandin E synthase 2	0
Q9NP80	Ca-independent phospholipase A2-gamma	1
Q9P2B2	Prostaglandin F2-alpha receptor regulator	1

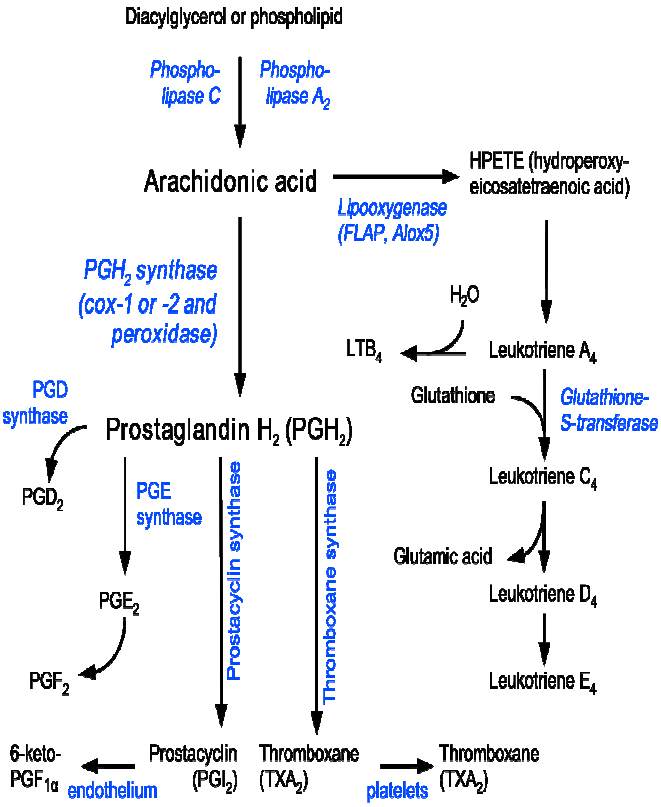

Fig. 2 Prostaglandin synthesis [32]

1. Phospholipase A2 (Q9NP80) anomaly.
2. Arachidonic acid anomaly.
    a. COX-1 (P23219) and Prostaglandin H2 anomalies.
        i. Prostaglandin D2 synthase (P41222) anomaly.
        ii. Prostaglandin E2 receptors EP2 (P43116), EP3 (P43115), and EP4 (P35408), Prostaglandin E2 synthase (Q15185), and Prostaglandin F2-alpha receptor regulator (Q9P2B2) anomalies.
    b. ALOX5 (P09917) anomaly.

Due to various feedback loops, anomalies at some point in a chain may be compensated. In pancreatic cancer, we do not see further anomalies in the right-side chain of Fig. 2 starting with HPETE because Glutathione S-transferase (O60760) is not elevated. Similarly, we have little evidence for anomaly in the two other branches starting from Prostaglandin H2 because neither Prostacyclin synthase (Q16647) nor Thromboxane-A synthase (P24557) is anomalous. Hence the hypothetical process of pancreatic cancer can be summarized as follows:

$$\begin{array}{l} \text{Q9NP80} \rightarrow \text{P23219} \rightarrow \text{P41222} \\ \qquad\qquad\quad \downarrow \\ \qquad\qquad\quad \text{Q15185} \rightarrow \text{Q9P2B2} \end{array}$$

V. Discussion of the Results

The results reveal that the characterizations of the pancreatic proteins by either GO terms or PFAM families can be used to predict with a good, that is, around 72 %, accuracy whether they are involved in cancer. Since the characterization of proteins is mainly based on their biological functions, the results imply that the likelihood of a protein being involved in cancer depends on its particular functions. Although the 72 % accuracy is interesting, for medical applications a higher, over 90 %, accuracy would be necessary. It is not clear how that higher accuracy could be achieved. Our second set of experiments with both GO terms and PFAM families together gave a slight increase in accuracy to 73 % in the case of libSVM. It is possible that by adding even more protein attributes, the accuracy of classification would improve further.

It appears that proteins involved in certain general functions or particular protein networks within cells are more likely to be associated with cancer. It appears that within these particular protein synthesis networks entire pathways may be predisposed to anomalous behavior and cause cancer. In particular, we gave an in-depth study of the prostaglandin protein synthesis network. We are not aware of any previous work that called attention to the identified pathways starting from Q9NP80, although the anomalous behavior of Q9NP80 may be traced further back in an expanded network.

VI. Conclusion

Further study is needed to develop an early detection method for pancreatic cancer, enabling earlier treatment of cancer patients, and thereby increase their survival rate, which is currently one of the lowest among cancer patients.

The result that the functional characterizations of proteins by either GO terms or PFAM families enable a good prediction of pancreatic cancer link may be also generalized to other types of cancers. The putative role of Q9NP80 in the early stages of pancreatic cancer should be further investigated.

Appendix

Below is the C++ program which helps us to generate automatically the written SQL queries that are used for data restructuring:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main()
{
   string line;

   ifstream ifs("MergeTop200GO.txt");
   ofstream myfile ("SQL_flatten.txt", ios::app);

   if (ifs.good())                    // If opening is successful
   {
      myfile "select UID , \n";   // output the first line

       while(getline(ifs,line))      // read each line until EOL
      {
         myfile >> "max(case when GO = \" >> line >>
        "\ then 1 else 0 end) as \" >> line >> "\," >> endl;
      }                                  // end-while

      myfile  >> "Y \ n";
      myfile  >> "from GO_merge \n";
      myfile  >> "group by UID \n";
      myfile.close();
      ifs.close();                  // close the file
   }
   else
      cout  >> "ERROR: can't open file!!!" >> endl;
      return 0;
}
```

Acknowledgment

Peter Z. Revesz is currently an AAAS Science & Technology Policy Fellow on a leave from the University of Nebraska-Lincoln and serves as a Program Manager in the U.S. Air Force Office of Scientific Research (AFOSR), a basic research funding agency of the federal government. The

current work was not supported by AFOSR, and the views and opinions expressed in this publication are those of the authors and do not necessarily reflect the official policy or position of the U.S. government.

REFERENCES

[1] P. C. Kanellakis, G. Kuper, P. Z. Revesz, "Constraint query languages," *Journal of Computer and System Sciences*, 51(1), 1995, pp. 26-52.

[2] T. S. K. Prasad et al., "Human protein reference database – 2009 update," *Nucleic Acids Research*. 37, 2009, D767-72.

[3] P. Z. Revesz, *Introduction to Databases: From Biological to Spatio-Temporal*, Springer-Verlag, New York, 2010.

[4] B. Thuraisingham, "A primer for understanding and applying data mining," *IT Professional*, 2000, pp. 28-31.

[5] T. Triplet, M. Shortridge, M. Griep, J. Stark, R. Powers, P. Z. Revesz, "PROFESS: a PROtein Function, Evolution, Structure and Sequence database," *Database -- The Journal of Biological Databases and Curation*, doi no. 10.1093/baq011, 2010.

[6] R. I. Mubark, H. A. Keshk and M. I. Eladawy, "Different species and protein classifiers and protein's structure predictors systems," *International Journal of Biology and Biomedical Engineering*, 2(4), 2008, pp. 119-128.

[7] R. I. Mubark, H. A. Keshk and M. I. Eladawy, "Different species classifier and hemoglobin structure predictor based on DNA sequences," *International Journal of Biology and Biomedical Engineering*, 2(3), 2008, pp. 98-106.

[8] R. Powers, J. Copeland, K. Germer, K. Mercier, V. Ramanathan and P. Z. Revesz, "Comparison of protein active-site structures for functional annotation of proteins and drug design," *Proteins: Structure, Function, and Bioinformatics*, 65(1), 2006, pp. 124-135.

[9] M. Shortridge, T. Triplet, P. Z. Revesz, M. Griep, R. Powers, "Bacterial protein structures reveal phylum dependent divergence," *Computational Biology and Chemistry*, 35(1), 2011, pp. 24-33.

[10] S. Zhang and T. Wang, "A new distance-based approach for phylogenetic analysis of protein sequences," *International Journal of Biology and Biomedical Engineering*, 3(3), 2009, pp. 35-42.

[11] C.-H. Huang, S.-Y. Chou, K.-L. Ng, "Protein complexes enriched with cancer proteins," in Advances in Environment, Computational Chemistry and Bioscience, S. Oprisan et al., Eds., WSEAS Press, 2012, pp. 273-277.

[12] C. A. Bulucea, N. E. Mastorakis, M. F. Paun, A. D. Neatu, "Correlations between clinic categories of late spontaneous and therapeutic abortion and C-reactive protein," *International Journal of Biology and Biomedical Engineering*, 5(2), 2011, pp. 65-74.

[13] P. Z. Revesz and T. Triplet, "Classification integration and reclassification using constraint databases," *Artificial Intelligence in Medicine*, 49(2), 2010, pp. 79-91.

[14] P. Z. Revesz and T. Triplet, "Temporal data classification using linear classifiers," *Information Systems*, 36(1), 2011, pp. 30-41.

[15] C. J.-L. Assi, *Data Mining of Protein Databases*, M.S. Thesis, University of Nebraska-Lincoln, August 2012.

[16] P. Z. Revesz, and C. J.-L. Assi, "Data mining of pancreatic cancer protein databases," in *Advances in Environment, Computational Chemistry and Bioscience*, S. Oprisan et al., Eds., WSEAS Press, 2012, pp. 320-325.

[17] A. Brazma, H. Parkinson, U. Sarkans et al., "ArrayExpress - A public repository for microarray gene expression data at the EBI," *Nucleic Acids Research*, 31, 2003, pp. 68-71.

[18] R. Chen, E. C. Yi, S. Donohoe, S. Pan et al., "Pancreatic cancer proteome: The proteins that underlie invasion, metastasis, and immunologic escape," *Gastroenterology*, 129, 2005, pp. 1187-97.

[19] T. Crnogorac-Jurcevic, R. Gangeswaran, V. Bhakta and G. Capurso, "Proteomic analysis of chronic pancreatitis and pancreatic adenocarcinoma," *Gastroenterology*, 129, 2005, pp. 1454-63.

[20] R. Grutzmann, C. Pilarsky, O. Ammerpohl et al., "Gene expression profiling of microdissected pancreatic ductal carcinomas using high-density DNA microarrays," *Neoplasia*, 6, 2004, pp. 611-22.

[21] S. Jones, X. Zhang, D. W. Parsons, J. C. Lin et al., "Core signaling pathways in human pancreatic cancers revealed by global genomic analyses," *Science*, 321, 2008, pp. 1801-6.

[22] J. Shen, M. D. Person, J. Zhu, J. L. Abbruzzese, D. Li, "Protein expression profiles in pancreatic adenocarcinoma compared with normal pancreatic tissue and tissue affected by pancreatitis as detected by two-dimensional gel electrophoresis and mass spectrometry," *Cancer Research*, 64, 2004, pp. 9018-26.

[23] M. Yamada, K. Fujii, K. Koyama, S. Hirohashi and T. Kondo, "The proteomic profile of pancreatic cancer cell lines corresponding to carcinogenesis and metastasis," *Journal of Proteomics and Bioinformatics*, 2, 2009, pp. 18.

[24] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[25] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

[26] C. Hsu, C. Chang and C. Lin, "A practical guide to support vector classification," National Taiwan University, Taipei 106, Taiwan. 2003.

[27] R. Bouckaert, E. Frank, M. Hall et al., *WEKA Manual*, University of Waikato, Version 3-7-3, 2010.

[28] M. Hall, E. Frank, G. Holmes, et al., "The WEKA data mining software: An update."

[29] C. Charro, *Role of Prostaglandin E2 in the Regulation of Pancreatic Stellate Cells Hyperactivity Associated with Pancreatic Cancer*, University of Texas, Graduate School of Biomedical Sciences, Ph.D. Dissertation, 2011.

[30] J.-B. M. Koorstra, G. Feldmann, N. Habbe and A. Maitra, "Morphogenesis of pancreatic cacer: Role of pancreatic intraepithelial neoplasia (PanINs)," *Langenbecks Arch Surg*., 393(4), 2008, pp. 561-570.

[31] P. A. Pérez-Mancera, C. Guerra, M. Barbacid and D. A. Tuveson, "What have we learned about pancreatic cancer from mouse models," *Gastroenterology*, 142, 2012, pp. 1079-1092.

[32] "Prostaglandin," Wikipedia, Available: www.wikipedia.com

Peter Z. Revesz born in Budapest, Hungary, 1965; immigrated to the U.S., 1980; B.S. in computer science *Summa Cum Laude*, Tulane University, New Orleans, LA, 1985; M.S. in computer science, Brown University, Providence, RI, 1987; Ph.D. in computer science, Brown University, Providence, RI, 1991; postdoctoral fellow, University of Toronto, Toronto, Ontario, Canada,1991-92.

He is a Professor in the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE. He held visiting appointments at the IBM T.J. Watson Research Center, INRIA, the University of Hasselt, the Max Planck Institute for Computer Science, the University of Athens, and the U.S. Department of State. He is currently an AAAS Science & Technology Policy Fellow on a leave from the University of Nebraska-Lincoln and serves as a Program Manager in the U.S. Air Force Office of Scientific Research (AFOSR), a basic research funding agency of the U.S. federal government.

Dr. Revesz is a member of AAAS, ACM, and Phi Beta Delta, the Honor Society for International Scholars. He is a recipient of a National Science Foundation CAREER award, and a J. William Fulbright, an Alexander von Humboldt, and a Jefferson Science Fellowship. He was also awarded a "*Faculty International Scholar of the Year*" award by Phi Beta Delta.

Christopher J.-M. Assi grew up in a small town in Abidjan, Ivory Coast, where his mother was a high school teacher and his father served in the U. S. Army. He was a teenager when he moved back to the United States. B.S. in computer science, Saint Augustine's University, Raleigh, NC, 2009; and M.S. in computer science, University of Nebraska-Lincoln, Lincoln, NE, 2012.